

PATENTTI- JA REKISTERIHALLITUS
NATIONAL BOARD OF PATENTS AND REGISTRATION

Helsinki 18.8.2000

ETUOIKEUSTODISTUS
PRIORITY DOCUMENT



Hakija
Applicant

Nokia Corporation
Helsinki

Patenttihakemus nro
Patent application no

19992834

Tekemispäivä
Filing date

30.12.1999

Kansainvälinen luokka
International class

H04L

Keksinnön nimitys
Title of invention

"Menetelmä ja laite datapakettien prosessoinnin ajoittamiseksi"

Täten todistetaan, että oheiset asiakirjat ovat tarkkoja jäljennöksiä patentti- ja rekisterihallitukselle alkuaan annetuista selityksestä, patenttivaatimuksista, tiivistelmästä ja piirustuksista.

This is to certify that the annexed documents are true copies of the description, claims, abstract and drawings originally filed with the Finnish Patent Office.


Pirjo Kaila
Tutkimussihteeri

Maksu 300,- mk
Fee 300,- FIM

Menetelmä ja laite datapakettien prosessoinnin ajoittamiseksi – Förfarande och anordning för att tidsplanera processering av datapaket

5 Esillä oleva keksintö koskee menetelmää ja laitetta datapakettien prosessoinnin ajoittamiseksi.

Tekniikan tausta

10 Perinteisesti soitettaessa puhelimella puhe on siirretty piirikytkentäisissä verkoissa, kuten puhelinverkossa (PSTN, Public Switched Telephone Network). Puhelimella soitettaessa digitaalisessa piirikytkentäisessä verkossa, jokaiselle puhelulle muodostetaan 64 kbps (kilo bits per second) kiinteä yhteys. Yhteyden vakiokaista, 64 kbps, johtuu analogisen puheen näytteenotossa tarvittavasta bittinopeudesta käytettäessä 8-bittistä pulssikoodimodulaatiota (PCM, Pulse Code
15 Modulation) 8-kHz:n näytteenottotaajuudella, mikä menettely mahdollistaa analogisen 300-3400 Hz:n puheen lähettämisen digitaalisessa muodossa.

Edellä kuvattu nykyisin yleisesti käytössä oleva digitaalinen puhelinverkko on kuitenkin hyvin tehoton ja käyttää siten paljon verkon resursseja. Puhelinverkossa
20 yhteyden kaista on varattuna myös silloin, kun yhteyttä ei aktiivisesti käytetä eli kumpikaan yhteyden osapuoli ei siirrä informatiota yhteyttä pitkin. Tällainen staattisen kaistan käyttö kuluttaa paljon tiedonsiirtoresursseja, minkä seurauksena käyttäjämäärien kasvaessa joudutaan investoimaan lisäkapasiteettiin. Lisäksi myös puhelinverkossa standardoidun tehottoman koodausmenettelyn (Coding
25 Scheme) takia tuhlataan kaistaa. Esimerkiksi G.729-koodaus suoriutuu näytteenotosta jopa niinkin alhaisella bittinopeudella kuin 8 kbps. Ongelmia edellä kuvatunlaisesta tehottomuudesta syntyy etenkin mannerten välisissä puheluissa, missä tiedonsiirtokapasiteetin kasvattaminen ei ole yhtä helppoa kuin muuten. Ongelma näkyy osittain myös puheluiden hinnoissa, kalliit investoinnit
30 kapasiteettiin on katettava korkeilla käyttömaksuilla.

Etenkin maiden välisiin yhteydenottoihin on staattisen kaistan varauksen sijasta alettu markkinoimaan niin sanottuja IP-puheluita (Internet Protocol Telephony). IP-

puhelussa puhe muunnetaan ensin analogisesta digitaaliseen muotoon, kompressoidaan ja lopuksi muunnetaan IP-paketeiksi, jotka kuljetetaan IP-verkon yli jakaen kaista muun IP-liikenteen kanssa. IP-puheluissa kaistaa voidaan käyttää huomattavasti tehokkaammin kuin staattisen kaistan varaavissa puheluissa, mikä näkyy myös puheluiden hinnoissa. Lisäksi voidaan käyttää myös uusia tehokkaampia koodausmenettelyjä, kuten esimerkiksi G.729-koodausta.

IP-puheluissa käyttäjä voi soittaa tavallisella puhelimella yhdyskäytävän kautta toiseen tavalliseen puhelimeen. Yhdyskäytävä toimittaa puhelun vastaanottajan yhdyskäytävään IP-pohjaisen tietoverkon, kuten esimerkiksi Internetin, välityksellä, josta puhelu ohjataan edelleen vastaanottajan paikallisen puhelinverkon kautta vastaanottajalle. Vastaajan yhdyskäytävässä puhelu yhdistyy takaisin yleiseen puhelinverkkoon. Toinen vaihtoehto on käyttäjän oleminen kiinteässä verkkoyhteydessä IP-pohjaiseen tietoverkkoon, esimerkiksi paikallisverkon välityksellä, jolloin hänen ei tarvitse lainkaan avata staattista puhekaistaa puhelinverkkoon vaan reititin, jonka takana hän on, voi reitittää hänen puhelunsa vastaanottajalle normaalin pakettipohjaisen tiedonsiirron tapaan. IP-puhelut perustuvat Internet-protokollaan, jonka avulla siirretään puhetta paketteina IP-verkon yli. Tämä tarkoittaa sitä, että IP-puheluita voidaan siirtää periaatteessa missä tahansa dataverkossa, joka käyttää IP-protokollaa, esimerkiksi Internetissä, Intraneteissä tai lähiverkoissa.

IP-puheluissa ongelmaksi muodostuu kuitenkin palvelun laatu (QoS, Quality of Service). IP-pakettien saapumisaikaa vastaanottajalle ei tiedetä ennen pakettien saapumista. IP-protokolla reitittää tietovuon pakettikohtaisesti, minkä ansiosta pakettien viive verkossa voi vaihdella suuresti ja pakettien järjestys voi vaihtua. Lisäksi paketit voivat hävitä esimerkiksi reitittimien puskureissa tapahtuvien tulevan datan ylivuotojen (over flow) seurauksena. Käyttämällä luotettavaa protokollaa kuten TCP (Transmission Control Protocol), voidaan tällaiset pakettihäviöt tunnistaa automaattisesti protokollan tasolla ja lähettää kadonneet paketit uudestaan. Kyseisenlaiset uudelleenlähetykset aiheuttaisivat kuitenkin edelleen vaihtelevaa viivettä pakettien matkatessa verkon halki, joten IP-puheluissa käytetään tavallisesti UDP-protokollaa (User Datagram Protocol), jossa

uudelleenlähetyksiä ei ole. Näinollen puheesta tulee helposti pätkittäistä ja sekavaa pakettien välisten taukojen kasvaessa vaikkei yhtään pakettia katoaisikaan matkalla.

- 5 Ratkaisua tähän ongelmaan on esitetty esimerkiksi julkaisussa, Ramjee R., Kurose K., Towsley D. 1994. Adaptive Playout Mechanism for Packetized Audio Applications in Wide-Area Networks., jossa tuleva pakettipohjainen puhetieto puskuroidaan ja useita paketteja käsittävän yhtenäisen puhepurskeen soittamisen aloittamista vastaanottajalle viivästetään. Viiveen pituuden määrittämisessä
- 10 hyödynnetään viimeksi tulleiden pakettien myöhästymisarvoista laskettua lyhyen aikavälin myöhästymistrendiä, eli myöhästymisarvoista laskettua liukuvaa keskiarvoa.

- Tällainen suora päästä-päähän (end-to-end) viivästämisen hallinta ei kuitenkaan
- 15 yleisesti riitä esimerkiksi interaktiivisen reaaliaikaisen datavirran laadun varmistamiseksi. Ei riitä pelkästään määrittää viivettä siten, että vain esimerkiksi yksi prosentti paketeista menetetään, kuten edellä kuvatussa mallissa. On myös tärkeää huomioida menetettyjen pakettien välinen korrelaatio, eli ns. häviökorrelaatio (loss correlation). Yhteyden laadun kannalta on suuri merkitys
 - 20 sillä menetetäänkö paketteja yksi siellä toinen toisaalla (ei häviökorrelaatiota) vai useita peräjäälkeen (suuri häviökorrelaatio). Häviökorrelaation merkitys kasvaa entisestään huomioitaessa lisäksi käytettävän koodekin ominaisuuksia, koska esimerkiksi VoIP- terminaalissa (Voice over Internet Protocol) käytetty koodekki, esimerkiksi G.723.1, voisi pystyä peittämään kahden peräkkäisen paketin
 - 25 menetyksen käyttämällä FEC-virheenkorjausta (Forward Error Correction) , mutta kolmen peräkkäisen paketin menetys saattaisi aiheuttaa kuultavissa olevan häiriön. Tällöin pitäisikin käytetyn menetelmän pystyä huomioimaan myös pakettien häviökorrelaatioita viivettä päättäessään. Tekniikan tasoa kuvastava menetelmä puskurin hallitsemiseksi ei kuitenkaan huomioi pakettien välisiä
 - 30 häviökorrelaatioita.

Keksinnön yhteenveto

Nyt on keksitty menetelmä ja laite datapakettien prosessoinnin ajoittamiseksi, joka parantaa esimerkiksi reaaliaikaisen pakettipohjaisen puheyhteyden puheenlaatua
 5 huomioimalla myös pakettien välisiä häviökorrelaatioita.

Keksinnön erään ensimmäisen aspektin mukaan on toteutettu laite datapakettien prosessoinnin ajoittamiseksi, käsittäen muistin laitteelle datapurskeen osana tulevan datapaketin tallentamiseksi, kellon ajankulun määrittämiseksi, ja
 10 prosessointivälineet muistista lähtevän datapaketin prosessoimiseksi, tunnettu siitä, että laite käsittää lisäksi laskentavälineet sellaisen arvon laskemiseksi toistoviiveelle, jolla n :stä ajallisesti viimeisestä datapaketista vain m kappaletta olisi jäänyt vastaanottamatta, jos kyseiset datapaketit käsittävien datapurskeiden prosessoinnin aloittamista olisi viivästytetty mainitun toistoviiveen verran, missä n
 15 ja m ovat luonnollisia lukuja, ja siirtovälineet pakettien siirtämiseksi muistista prosessointivälineille kelloilta saatavan vasteen pohjalta mainitun toistoviiveen arvon saavuttamisesta siitä kun datapaketti vastaanotettiin.

Keksinnön erään toisen aspektin mukaan on toteutettu menetelmä datapakettien
 20 prosessoinnin ajoittamiseksi, jossa menetelmässä vastaanotetaan datapurskeen osana oleva datapaketti, tallennetaan vastaanotettu datapaketti muistiin, otetaan datapaketti muistista toistoviiveen jälkeen siitä kun datapaketti vastaanotettiin, tunnettu siitä, että lasketaan toistoviiveelle arvo, jolla toistoviiveen arvolla n :stä ajallisesti viimeisestä datapaketista vain m kappaletta olisi jäänyt
 25 vastaanottamatta, jos kyseiset datapaketit käsittävien datapurskeiden prosessoinnin aloittamista olisi viivästytetty mainitun toistoviiveen verran, missä n ja m ovat luonnollisia lukuja, ja siirretään datapaketti muistista prosessointivälineille kelloilta saadun vasteen pohjalta mainitun toistoviiveen arvon saavuttamisesta siitä kun datapaketti vastaanotettiin.

30

Datapurskeella tarkoitetaan tässä yhteydessä purskeisen informaation, kuten puheen tai videon, yhtäjaksoista tauotonta lähettämistä. Datapurskeen kesto on siis esimerkiksi sen pituinen kuin lähettäjä vaikkapa puhuu yhtäjaksoisesti, ja siten

lähettäjän pitäessä tauon puhumisessaan myös datapurske loppuu. Erilaisia datapurskeita ovat esimerkiksi puhepurske, jossa lähetetään purskeista puheinformaatiota, ja videopurske, jossa lähetetään purskeista videokuvaa.

Datapurskeeseen voi siten kuulua suurikin määrä datapaketteja, joilla

- 5 datapaketeilla käsitetään tässä yhteydessä ensisijaisesti analogisesta informaatiosta, kuten puhe ja kuva, otettua digitaalista näytettä. Datakehyksellä taas käsitetään seuraavassa esimerkiksi IP-pohjaisessa tietoverkossa lähetettäessä datapaketin/datapakettien ympärille asetettavan otsikkokentän ja datapaketin/datapakettien muodostamaa yhtenäistä kokonaisuutta.

10

Keksinnön mukainen laite ja menetelmä pohjautuvat olettamuksiin peräkkäin menetettyjen pakettien lukumäärän rajoituksesta ja maksimi toistoviiveestä datapurskeen ensimmäisen datapaketin prosessoinnin aloittamisessa. Peräkkäin menetettyjen pakettien lukumääräraja riippuu esimerkiksi käytetyn koodekin ominaisuuksista. Jos käytetty koodekki pystyy korjaamaan kahden peräkkäin

15 menetetyn datapaketin menetyksen käytetään peräkkäin menetettyjen datapakettien maksimilukumääränä ℓ_{\max} arvoa kaksi, mikäli muita asiaan vaikuttavia tekijöitä ei ole. Toistoviive d taas asetetaan siten, että peräkkäisten

menetettyjen datapakettien lukumäärä ℓ , n :stä viimeksi vastaanotetusta

- 20 datapaketista käsin tarkasteltuna, olisi enintään peräkkäin menetettyjen datapakettien maksimilukumäärä ℓ_{\max} , jos vain toistoviiveen maksimiarvo d_{\max} sallii näin suuren toistoviiveen arvon.

Yleisesti voidaan sanoa peräkkäisten menetettyjen pakettien lukumäärän

- 25 pienenevän toistoviiveen arvon kasvaessa. Ääritapauksessa voitaisiin ajatella toistoviiveen olevan niin suuri, että koko datapurske ehdittäisiin vastaanottamaan ennen ensimmäisen datapaketin prosessoinnin aloittamista toistoviiveen jälkeen kyseisen datapaketin vastaanottamisesta, tällöin yhtään datapakettia ei menetettäisi myöhästymisen takia. Näin suuri toistoviiveen arvo kuitenkin
- 30 romuttaisi yhteyden kaksisuuntaisen (full duplex), reaaliaikaisuuden ja interaktiivisuuden. Suurilla toistoviiveen arvoilla yhteyden koettu laatu (QoS) heikkenee esimerkiksi yhteydessä olevien osapuolien lähettämien informaatioiden,

kuten puheen, mennessä päällekkäin. Toistoviiveen maksimiarvon määrittäminen onkin monitavoiteoptimointitehtävä, tavoitteena toistoviiveen ja peräkkäin menetettyjen datapakettien lukumäärän minimointi, jonka ratkaisu yleensä muuttuu verkon olosuhteiden muuttuessa.

5

Eli toisin sanottuna keksinnön mukaisessa menetelmässä lasketaan toistoviiveelle d arvo, joka on pienin mahdollinen arvo, jolle pätee $\ell \leq \ell_B^{\max}$ kuitenkin siten, että toistoviiveelle pätee $d \leq d_{\max}$, eli jolle arvolle pätee, että se on pienempi tai yhtä pieni kuin toistoviiveen maksimiarvo, tai menetettyjen peräkkäisten datapakettien lukumäärä on pienempi tai yhtä pieni kuin menetettyjen peräkkäisten datapakettien maksimiarvo, jos tämä voidaan saavuttaa toistoviiveellä, joka on pienempi kuin toistoviiveen maksimiarvo.

10

Seuraavassa datapaketin viivästymisajalla tarkoitetaan datapurskeen ensimmäisen datapaketin vastaanottajan laitteelle saapumisesta laskettua datapaketin teoreettista saapumisajasta myöhästymistä. Jos k on datapurskeen ensimmäisen datapaketin saapumisaika ja v on lähettäjän käyttämä näytteenottoväli, saadaan datapurskeen n :nnen datapaketin teoreettinen saapumisaika laskettua yhtälöstä saapumisaika $= k + n * v$, mihin arvoon toteutunutta saapumisaikaa sitten verrataan viivästymisajan laskemiseksi. Käytettäessä RTP-protokollaa, jolloin datakehysten RTP-otsikkokenttä sisältää sekä datapaketin aikaleiman että järjestysnumeron, teoreettinen saapumisaika voidaan laskea vertaamalla datapurskeen ensimmäisen datapaketin ja laskennan kohteena olevan datapaketin aikaleimoja toisiinsa. Tämä lähestymistapa olisikin ehdoton mikäli jostain syystä lähettäjän käyttämä näytteenottoväli ei olisi vakio datapurskeen sisällä.

20

25

30

Keksinnön eräässä edullisessa suoritusmuodossa laitetta ja menetelmää käytetään verkkopuhelussa datapurskeiden datapakettien viiveiden hallinassa. Keksinnön mukaisella menetelmällä viivästetään datapurskeen ensimmäisen datapaketin prosessoinnin aloittamista sen verran, että jo saapuneiden edellisten datapurskeiden datapakettien viivästymisajoista arvioiden n :stä viimeisestä jo

saapuneesta datapaketista olisi toistoviiveen puitteissa maksimissaan jäänyt saapumatta m:n peräkkäisen datapaketin pituinen ryhmä. Keksinnön mukainen laite voi olla erillinen verkkopuhelin, verkkokuvapuhelin, langaton puhelin tai muu vastaava tai se voi hyödyntää tietokoneen prosessoria, mikrofonia, kaiuttimia, videokorttia ynnä muita sellaisia, jolloin sen käytännön toteuttamisessa ei välttämättä tarvitsisi erillistä laitetta vaan se voitaisiin toteuttaa ohjelmallisesti hyödyntämällä nykyaikaisista tietokoneista yleisesti löytyviä laitteita.

Keksinnöllä aikaansaadaan suhteellisen helppo ja yksinkertainen menetelmä ja laite datapakettien prosessoinnin ajoittamiseksi. Keksintö on erityisen käyttökelpoinen reaaliaikaisessa interaktiivisessa yhteydenpidossa tietoverkon, kuten esimerkiksi Internetin, kautta. Keksinnöllä pystytään aikaisempaa tehokkaammin mukautumaan vastaanottajapäässä stokastisten verkko-olosuhteiden aiheuttamien eri pituisten viivästymisten aiheuttamiin ongelmiin, ja siten parantamaan koettua yhteydenlaatua (QoS, Quality of Service). Tämä aikaansaadaan, aikaisemmasta poiketen, huomioimalla pelkän datapakettien viivästymisten lisäksi myös saapumattomien datapakettien välisiä häviökorrelaatioita, jotka saattavat vaikuttaa merkittävästi koettuun yhteyden laatuun.

Keksintöä selostetaan seuraavassa yksityiskohtaisesti viittaamalla oheisiin piirustuksiin, joissa

kuviossa 1 on esitetty yleisesti lohkokaaviona keksinnön mukainen

toimintaympäristö,

kuviossa 2 on esitetty yleisesti lohkokaaviona eräs keksinnön mukaisen edullisen toimintaympäristön arkkitehtuuri,

kuviossa 3 on esitetty kaavio keksinnön mukaisen laitteen eräästä edullisesta toimintamuodosta toimintaympäristössään,

Kuviossa 4 on esitetty yleisesti IP-pohjaisissa tietoverkoissa reaaliaikaisen datan tiedonsiirrossa käytettävä datakehys,

kuviossa 5 on esitetty lohkokaaviona kuvaus keksinnön mukaisesta laitteesta datapakettien prosessoinnin ajoittamiseksi,

kuviossa 6 on kuvattu vuokaaviolla yhtä keksinnön mukaisen menetelmän edullista toimintamuotoa, ja kuviossa 7 on esitetty havaintoesimerkki toistoviiveen ja menetettyjen peräkkäisten datapakettien muodostaman jonon pituuden välisestä suhteesta.

5

Kuviossa 1 on esitetty yleisesti keksinnön erään suoritusmuodon mukaisen laitteen eräs edullinen toimintaympäristö lohkokaaavana. Toimintaympäristö käsittää reaaliaikasovelluksia, kuten esimerkiksi VoIP:ta tai pakettipohjaista videota, käyttävän päätelaitteen 10, joka päätelaite voi olla esimerkiksi VoIP-puhelin, kuvapuhelin, tietokoneen reaaliaikainen interaktiivinen

10

videotapaamissovellus tms. Päätelaite 10 on sovittimen 11, joka voi koostua esimerkiksi puhelinverkossa (PSTN, Public Switched Telephone Network) sijaitsevasta yhdyskäytävästä (gateway), välityksellä yhteydessä IP-pohjaiseen verkkoon 12 (IP cloud, Internet Protocol), kuten esimerkiksi Internetiin. Internet on

15

yksinkertaistetusti eri verkoista muodostunut verkko, joka tukee TCP/IP-pohjaisia sovelluksia, kuten esimerkiksi WWW:tä (World Wide Web), SMTP (Simple Mail Transfer Protocol) email:iä tai FTP:tä (File Transfer Protocol). Internetin

muodostavat osaverkot voivat olla esimerkiksi ATM- 13 (Asynchronous Transfer Mode), PPP/SDH- 14 (Point-to-Point Protocol / Synchronous Digital Hierarchy),

20

Ethernet- 15 (kts. esimerkiksi standardi IEEE 802.3), X25- 16 (ITU-T:n standardi), FR- 17 (Frame Relay), tai FDDI-pohjaisia 18 (Fiber Distributed Data Interface) verkkoja. Internetiin on kytkeytynyt käyttäjiä, jotka käyttävät esimerkiksi ftp- ja email-pohjaisia sovelluksia 19 tai esimerkiksi telnet- ja http-pohjaisia (Hypertext Transfer Protocol) sovelluksia 20. On ilmeistä, ettei keksinnön mukainen

25

toimintaympäristö rajoitu edellä esitettyyn vaan todellisuudessa Internet koostuu paljon suuremmasta määrästä käyttäjiä ja verkkoja.

Kuviossa 2 on esitetty yleisesti lohkokaaavana eräs keksinnön mukaisen laitteen edullinen toimintaympäristöarkkitehtuuri (VoFR-arkkitehtuuri)(Voice over Frame Relay), joka toimintaympäristö käsittää reaaliaikasovelluksia hyödyntäviä päätelaitteita 21 ja päätelaitteiden väliseen reaaliaikaiseen tiedonsiirtoon viiveitä aiheuttavia reitittimiä 29. Päätelaitteet 21 on kytketty ATM-pohjaiseen FR-verkkoon 22, joko puhelinverkon kautta (PSTN) 23 yhdyskäytävää 24 pitkin tai

30

päätelaitteiden ollessa suorassa verkkoyhteydessä PBX:n 25 (private branch exchange) kautta VFRAD-laitteeseen 26 (Voice Frame Delay Access Device) ja sieltä edelleen FR-verkkoon. VFRAD mahdollistaa datan, äänen ja faxin multipleksoimisen/demultipleksoimisen eri lähteiden ja sovellusten ja yhden verkkoliitäntäpisteen (network access point) välillä. VFRAD:n toisella puolella voi olla myös paikallisverkon reititin 27 ja reitittimen 27 takana useita päätelaitteita 28.

Kuviossa 3 on kaaviolla havainnollistettu keksinnön mukaisen laitteen erästä edullista käyttömuotoa toimintaympäristössään. Toimintaympäristö koostuu kahdesta toisiinsa verkon yli reaaliaikaisessa interaktiivisessa yhteydessä olevasta osapuolesta - lähettäjistä 31 ja vastaanottajasta 39. Koska on kyseessä interaktiivinen yhteys kahden osapuolen välillä on selvää, että osapuolten roolit lähettäjänä ja vastaanottajana saattavat vaihtua yhteyden aikana. Lähettäjä 31 on reaaliaikaisessa interaktiivisessa tiedonsiirtoyhteydessä, kuten esimerkiksi VoIP-yhteydessä, vastaanottajan 39 kanssa. Lähettäjän lähettämästä analogisesta informaatiosta, esimerkiksi puheesta, otetaan näytteitä sopivin vakioväliajoin näytteenottimella 32, joka on esimerkiksi mikrofoni, digitaalinen videokamera tai muu vastaava A/D muuntimen käsittävä näytteenotin, analogisen tiedon muuttamiseksi digitaaliseen muotoon. Saadut näytteet varastoidaan lähetyspuskuriin 33 odottamaan lähetyspakkausta. Lähetyspuskurista näytteet ohjataan kooderille 34, joka koodaa ne verkossa siirrettävään muotoon, esimerkiksi kehystää ne. Kooderilta datapaketti lähetetään verkkoon 35 kuljetettavaksi vastaanottajalle 39. Datapaketien kulkua verkossa hidastuttavat lähinnä reitittimissä 29 (kuvassa 2) jonottamiset ja fyysisen etenemisen (propagation) viemä aika. Fyysiseen etenemiseen käytetty aika voi myös olla eri pituinen eri datapaketeille, tai tarkemmin datakehyksille koska datapaketit on tässä vaiheessa asetettu kehyksen sisään, johtuen esimerkiksi jonkun reitittimen reititystaulukon muuttumisesta kesken datapurskeen lähetyksen, joka näin ohjaa muutoksen jälkeen tulleet datapaketit eri reittiä pitkin kuin ennen reititystaulukon päivittymistä. Verkosta datapaketit siirtyvät jitter-puskuriin 36 (jitter-buffer) IP- ja UDP-protokollapinojen kautta, jotka poistavat datakehuksesta ko. protokollia vastaavat kentät, odottamaan prosessointiaan ja siirtämistään dekodeerille 37. Vastaanottajan päätelaite ottaa puskurista 36 näytteitä määrättyjen kriteerien

täyttyessä ja siirtää ne dekooderille 37, missä datakehys puretaan, ja sieltä edelleen soittajalle 38 (player), kuten esimerkiksi tietokoneen puheensoittosovellus, videonkäsittelysovellus, verkkopuhelimen D/A muuntimen ja kaiuttimen yhdistelmä tai muu vastaava D/A muuntimen käsittävä sovellus,

5 muunnettaviksi digitaalisesta takaisin analogiseen muotoon, prosessoitaviksi ja esitettäviksi siten vastaanottajalle.

Kuviossa 4 on esitetty yleisesti IP-pohjaisissa tietoverkoissa reaaliaikaisen datan tiedonsiirrossa käytettävä datakehys 41. Datakehys käsittää IP-otsikkokentän 42,

10 joka käsittää yleisesti lähinnä tietoa lähettäjän ja vastaanottajan IP-osoitteista. IP-otsikkokentän jälkeen seuraa esimerkiksi UDP-otsikkokenttä 43, UDP-pohjaisen tiedonsiirron olleessa kysymyksessä, mitä esimerkiksi IP-verkossa siirrettävät puhe- ja videoinformaatio yleisesti käyttävät. UDP-otsikkokenttä käsittää tietoa lähettäjän ja vastaanottajan porttien numeroista ja haluttaessa kenttään voidaan

15 lisätä ylimääräinen datakentän tarkistussumma. UDP-otsikkokenttää seuraa RTP-otsikkokenttä 44, joka käsittää tietoa esimerkiksi siitä mille vastaanottajan sovellukselle kyseinen datapaketti on menossa ja tietoa siitä miten kyseinen datapaketti asettuu samalle sovellukselle siirrettävään tietovirtaan, eli datapaketin järjestysnumeron. RTP-otsikkokenttää seuraa varsinainen tietoverkon yli

20 lähetettävä datapaketti 45 tai useita datapaketteja.

Kuviossa 5 on esitetty lohkokaaviona kuvaus keksinnön mukaisesta laitteesta datapakettien prosessoinnin ajoittamiseksi. Laite käsittää tietoliikenneyhteyden 51 IP-pohjaiseen tietoverkkoon 12, joka tietoliikenneyhteys voi olla toteutettu

25 esimerkiksi yleisen puhelinverkon ja yhteyskäytävän välityksellä, suoraan paikallisverkon kautta tai langattoman yhteyden kautta tietoverkkoon. Tietoliikenneyhteyttä 51 pitkin laitteelle siirretään tietoa datakehysten 41 sisällä tulevissa datapaketeissa 45. Datapaketeissa tuleva tieto vastaanotetaan laitetta ohjaavassa yksikössä 52 (MCU, Master Controlling Unit), joka on esimerkiksi

30 mikroprosessori. Laitetta ohjaava yksikkö on järjestetty muistiin 53 tallennetun ohjelman perusteella tallentamaan datakehyksessä tulleen datapaketin tai datapaketit muistiin 53. Kuviossa on alempana esitetty laitteenohjausosan 52 fyysinen lohkokaavio ja kuviossa on ylempänä katkoviivan sisällä esitetty

laitteenohjausosan 52 toiminnallinen lohkokaavio. Laitetta ohjaava yksikkö käsittää lisäksi kellon 54, esimerkiksi mikroprosessorin kello, ajankulun määrittämiseksi ja muistiin 53 tallennetuilla ohjelmilla toteutetut toiminnot datapakettien myöhästymisen laskemiseksi 55 lähetysajan ja saapumisajan erotuksesta ja toistoviiveen

- 5 laskemiseksi 56 n:n viimeisen datapakettien viivästymisarvojen perusteella. Laitetta ohjaava yksikkö on järjestetty muistiin 53 tallennetun ohjelman perusteella purkamaan datakehysten dekooderi-sovelluksella kellosta saatavan vasteen pohjalta. Lisäksi muistiin on tallennettu siirtotoiminto 57 datapakettien siirtämiseksi prosessoitavaksi laitetta ohjaavaan yksikköön ja esitettäväksi käyttäjälle soittajalla
- 10 38 kellosta saatavan vasteen pohjalta määrätyn aikarajan saavuttamisesta.

Kuviossa 6 on kuvattu vuokaaviolla yhtä keksinnön mukaisen menetelmän edullista toimintamuotoa, missä menetelmässä laitteelle tulee tietoverkosta tietoliikenneyhteyttä pitkin datakehys 41 (vaihe 61). Laitteessa datakehys

- 15 puretaan, eli datakehysten käsittämät otsikkokentät poistetaan ja itse datapaketit otetaan talteen (vaihe 62). Datapakettien saatavien tietojen pohjalta lasketaan kyseisen datapakettien viivästymisaika, eli se toistoviive jolla paketin sisältämä ääninäyte olisi voitu käyttää hyväksi, ja tämä viivästymisaika talletetaan muistiin, jossa on n:n viimeksi tulleen datapakettien viivästymisaikat tallennettuina (vaihe 63),
- 20 jossa n on luonnollinen luku. Lasketaan päivitetyn n:n viimeisen viivästymisaikan perusteella uusi arvo toistoviiveelle, jolla toistoviiveen arvolla n:stä viimeksi saapuneesta datapakettien maksimissaan m peräkkäisen datapakettien jono olisi jäänyt saapumatta ajallaan viivästytettäessä datapurskeen ensimmäisen datapakettien prosessoinnin aloittamista toistoviiveen verran sen saapumisesta ja
- 25 prosessoitaessa datapurskeen ensimmäistä datapakettia seuraavia datapaketteja näytteenottovälein ensimmäisen datapakettien prosessoinnin jälkeen (vaihe 64). Edellisessä mainittu arvo m voi edullisesti olla esimerkiksi 2 ja mainittu arvo n voi edullisesti olla esimerkiksi luokkaa muutama kymmenen.

- 30 Siinä erikoistapauksessa, jossa datapaketti on koko kyseisen yhteyden ensimmäinen datapaketti, jolloin ei ole olemassa tietoa yhteyden aikaisempien datapakettien viivästymisajoista, asetetaan toistoviiveen arvoksi alkuarvausarvo, joka on edullisesti esimerkiksi edellisellä kerralla laitteella käytetty viimeinen

toistoviiveenarvo tai jonkinlainen muu arvausarvo, esimerkiksi tehdasasetusten perusteella.

Uusien arvojen laskemistoimenpiteiden jälkeen tunnistetaan vielä onko datapaketti uuden datapurskeen, kuten esimerkiksi puhepurskeen tai videopurskeen, ensimmäinen datapaketti vai onko kyseessä datapurskeen ensimmäistä datapakettia seuraava datapaketti (vaihe 65). Tämän jälkeen tallennetaan kyseinen datapaketti laitteen muistiin odottamaan kyseisen datapaketin prosessoitavaksi siirtämistä (vaihe 66). Tietysti siinä tapauksessa, että kyseisen datapaketin prosessointiaika on jo ohitettu ei datapakettia talleteta, mutta datapaketin viivästymisaika kylläkin.

Edellä kuvatut vaiheet 63-65 voidaan myös suorittaa edullisesti edellä kuvatusta poikkeavassa järjestyksessä ilman hankaluuksia ja siten myös nämä toteutusmuodot kuuluvat keksinnön mukaisen menetelmän piiriin.

Mikäli datapaketti oli uuden datapurskeen ensimmäinen datapaketti asetetaan kellon tarkkailemaksi viiveajaksi mainittu laskettu uusi toistoviiveen arvo kyseisen datapaketin laitteelle saapumisajasta laskien (vaihe 67). Mikäli datapaketti ei ollut datapurskeen ensimmäinen datapaketti asetetaan kellon tarkkailemaksi viiveajaksi lähettäjän näytteenottoväli, joka yleensä, mutta ei välttämättä, on vakio, kyseistä datapakettia edeltävän datapaketin prosessoitavaksi siirtämisestä laskennan aloittaen (vaihe 68). Esimerkiksi puhepurskeessa purskeen ensimmäistä datapakettia voidaan viivästyttää niin pitkään kuin halutaan kuullun puheen laadun kärsimättä, suurilla viivästyksillä kuitenkin menetettäisiin reaaliaikainen kaksisuuntaisuus, joten viivearvolle on olemassa maksimiarvo. Datapurskeen ensimmäistä datapakettia seuraavia datapaketteja ei voida kuitenkaan enää viivästyttää vaan ne täytyy prosessoida, tai olla prosessoimatta siinä tapauksessa etteivät ne ole saapuneet määrättyyn aikaan mennessä, määrättyin väliajoin toisistaan Tämä väliaika on yhtä suuri kuin näytteenottoaikaväli lähettäjän päässä yhteyttä. Esimerkiksi puheen tapauksessa muiden kuin ensimmäisen datapaketin viivästyttäminen vakioviiveestä poiketen huomattaisiin puheen "laahaamisena" vastaanottajalle soitettaessa tai vastaavasti liian nopea datapakettien soittaminen

havaittaisiin normaalia nopeampana ja eri kuuloisena puheena. Kummassakin tilanteessa vastaanottajan tuntema yhteyden laatu (Quality of Service) huononisi viiveen erotessa näyttenottovälistä. Edellä kuvatuslaisen efektin voi myös helposti todeta esimerkiksi videonauhurilla hidastuttamalla videonauhalle tallennetun

5 tiedon prosessointia videonauhurissa painamalla hidastusvalitsinta tai vastaavasti nopeuttamalla videonauhan informaation prosessointia.

Viiveaikojen asettamisen jälkeen odotetaan kunnes asetettu viiveaika täyttyy.

Datapurskeen ensimmäisen datapakettien tapauksessa odotetaan, että toistoviive

10 kyseisen datapakettien saapumisesta laitteelle saavutetaan (vaihe 69).

Datapurskeen muiden datapakettien tapauksessa odotetaan, että näyttenottovälin mittainen aika kyseistä datapakettia edeltävän datapakettien siirtämisestä, tai siitä kuin ajallisesti edellinen datapaketti olisi pitänyt siirtää mutta jäi esimerkiksi saapumatta siirtoaikansa puitteissa eikä siten kuitenkaan tullut siirrettyä,

15 prosessoitavaksi saavutetaan (vaihe 70). Toistoviiveen, datapakettien ollessa datapurskeen ensimmäinen datapaketti, tai näytteenottovälin, datapakettien ollessa muu kuin datapurskeen ensimmäinen datapaketti, mennessä umpeen siirretään kyseinen datapaketti prosessoitavaksi esimerkiksi soittajalle (player) 38 tai jollekin muulle sovellukselle tai välineelle (vaihe 71).

20 Edellä kuvattu menetelmä voitaisiin myös kuvata kaksiosaisena, jolloin ensimmäisen osan muodostaisivat vaiheet 61-66 ja toinen osa kävisi kellolta saatavan vasteen pohjalta hakemasta puskurista datapakettien, mikäli datapaketti olisi ehtinyt saapua. Menetelmässä olisi siis osa joka veisi tiedot ja datan puskuriiin

25 ja muistiin ja osa joka hakisi datan ja käyttäisi muistiin talletettuja tietoja viiveiden laskennassa.

Kuviossa 7 on esitetty havaintoesimerkki toistoviiveen ja menetettyjen peräkkäisten datapakettien muodostaman jonon pituuden välisestä suhteesta.

30 Kuviossa pystyakselilla on esitetty datapurskeen ensimmäisen datapakettien, ja siten koko datapurskeen, viivästämisessä käytettävän toistoviiveen arvo millisekunneissa ja vaaka-akselilla on ilmoitettu datapurskeen datapakettien järjestysnumerot. Jatkuva viiva (merkitty "tr1") kuvaa kunkin datapakettien

myöhästymistä datapurskeen ensimmäisen datapaketin ja näytteenottoaikavälin perusteella lasketusta teoreettisesta saapumisajasta. Katkoviivat (30 ja 40) kuvaavat, toistoviiveiden arvoilla 30 ja 40 millisekuntia, sitä ehtivätkö datapaketit prosessoitaviksi kyseisten toistoviiveiden arvojen puitteissa, jos datapaketti saapui

5 katkoviivan yläpuolella se ei olisi ehtinyt saapua prosessoitavaksi viiveajan puitteissa ja jos se saapui katkoviivan alapuolella se olisi ehtinyt saapua prosessoitavaksi viiveajan puitteissa. Kuten kuvioista käy ilmi pisin viiveajan puitteissa saapumatta jäänyt peräkkäisten datapakettien muodostama jono toistoviiveen arvolle 30 millisekuntia sisältää kahdeksan datapakettia

10 (järjestysnumeroiden 540-560 välillä). Toistoviiveen arvolle 40 millisekuntia pisin viiveajan puitteissa peräkkäisten saapumatta jääneiden datapakettien muodostama jono oli enää vain kahden paketin pituinen. Kahden peräkkäisen datapaketin menetys voitaisiin saada korjattua käytettävällä koodekillä ja siten toistoviiveen arvolla 40 millisekuntia yhteydessä ei havaittaisi datapakettien

15 häviöitä. Tässä tapauksessa kuviossa esitettyjen viivästymisaikojen perusteella toistoviiveen arvoksi voitaisiinkin edullisesti valita 40 millisekuntia.

Tässä on esitetty keksinnön toteutusta ja suoritustapoja esimerkkien avulla. Alan ammattimiehelle on ilmeistä, ettei keksintö rajoitu edellä esitettyjen

20 suoritustapojen yksityiskohtiin ja että keksintö voidaan toteuttaa muussakin muodossa poikkeamatta keksinnön tunnusmerkeistä. Esitettyjä suoritustapoja tulisi pitää valaisevina, muttei rajoittavina. Siten keksinnön toteutus- ja käyttömahdollisuuksia rajoittavatkin ainoastaan oheistetut patenttivaatimukset. Täten vaatimusten määrittelemät erilaiset keksinnön toteutusvaihtoehdot, myös

25 ekvivalenttiset toteutukset kuuluvat keksinnön piiriin.

Patenttivaatimukset

1. Laite datapakettien prosessoinnin ajoittamiseksi, käsittäen muistin (53) laitteelle datapurskeen osana tulevan datapaketin tallentamiseksi, kellon (54) ajankulun määrittämiseksi, ja prosessointivälineet (38) muistista
5 lähtevän datapaketin prosessoimiseksi, **tunnettu** siitä, että laite käsittää lisäksi laskentavälineet (56) sellaisen arvon laskemiseksi toistoviiveelle, jolla n:stä ajallisesti viimeisestä datapaketista vain m kappaletta olisi jäänyt vastaanottamatta, jos kyseiset datapaketit käsittävien datapurskeiden prosessoinnin aloittamista olisi viivästytetty mainitun toistoviiveen verran, missä n
10 ja m ovat luonnollisia lukuja, ja siirtovälineet (57) pakettien siirtämiseksi muistista prosessointivälineille (38) kelloilta (54) saatavan vasteen pohjalta mainitun toistoviiveen arvon saavuttamisesta siitä kun datapaketti vastaanotettiin.
2. Patenttivaatimuksen 1 mukainen laite, **tunnettu** siitä, että lisäksi laite käsittää
15 toistoviiveelle määritetyn maksimiarvon ja mainitun toistoviiveen arvon ollessa suurempi kuin toistoviiveelle määritetty maksimiarvo on laite järjestetty käyttämään toistoviiveen arvona toistoviiveen maksimiarvoa.
3. Patenttivaatimuksen 1 mukainen laite, **tunnettu** siitä, että mainittu datapaketti
20 on datapurskeen ensimmäinen datapaketti.
4. Patenttivaatimuksen 3 mukainen laite, **tunnettu** siitä, että mainitut siirtovälineet (57) on järjestetty siirtämään ensimmäistä datapakettia seuraavia samaan datapurskeeseen kuuluvia vastaanotettuja datapaketteja muistista viiveen jälkeen
25 siitä kun ajallisesti edellinen datapaketti otettiin muistista.
5. Patenttivaatimuksen 4 mukainen laite, **tunnettu** siitä, että mainittu viive on näytteenottoväli.
- 30 6. Patenttivaatimuksen 1 mukainen laite, **tunnettu** siitä, että mainittu datapurske käsittää reaaliaikaista interaktiivista dataa.

7. Patenttivaatimuksen 6 mukainen laite, **tunnettu** siitä, että datapurske käsittää yhtä seuraavista: packet video over IP, Voice over IP, audio/video streaming over IP.

5 8. Patenttivaatimuksen 1 mukainen laite, **tunnettu** siitä, että mainitun datapurskeen kesto on yhtäpitkä aika kuin jona lähettäjä lähettää informaatiota tauottomasti.

9. Patenttivaatimuksen 1 mukainen laite, **tunnettu** siitä, että mainitut
10 datapaketit vastaanotetaan tietoverkosta reaaliaikaisen yhteyden kautta johonkin toiseen tietoverkon osapuoleen.

10. Patenttivaatimuksen 1 mukainen laite, **tunnettu** siitä, että mainittu laite on reaaliaikaista informaatiota hyödyntävän laitteen synkronointivirhepuskuri (jitter
15 buffer).

11. Menetelmä datapakettien prosessoinnin ajoittamiseksi, jossa menetelmässä

vastaanotetaan datapurskeen osana oleva datapaketti,
20 tallennetaan vastaanotettu datapaketti muistiin (53),
otetaan datapaketti muistista toistoviiveen jälkeen siitä kun datapaketti vastaanotettiin,
tunnettu siitä, että
lasketaan toistoviiveelle arvo, jolla toistoviiveen arvolla n :stä ajallisesti
25 viimeisestä datapaketista vain m kappaletta olisi jäänyt vastaanottamatta, jos kyseiset datapaketit käsittävien datapurskeiden prosessoinnin aloittamista olisi viivästytetty mainitun toistoviiveen verran, missä n ja m ovat luonnollisia lukuja, ja
siirretään datapaketti muistista prosessointivälineille (38) kelloilta (54) saadun vasteen pohjalta mainitun toistoviiveen arvon saavuttamisesta siitä kun datapaketti
30 vastaanotettiin.

12. Patenttivaatimuksen 11 mukainen menetelmä, **tunnettu** siitä, että lisäksi toistoviiveelle on määritetty maksimiarvo ja mainitun toistoviiveen arvon ollessa suurempi kuin toistoviiveelle määritetty maksimiarvo käytetään toistoviiveen arvona toistoviiveen maksimiarvoa.

5

13. Patenttivaatimuksen 11 mukainen menetelmä, **tunnettu** siitä, että mainittu datapaketti on datapurskeen ensimmäinen datapaketti.

14. Patenttivaatimuksen 13 mukainen menetelmä, **tunnettu** siitä, että

10

otetaan vastaanotettuja samaan datapurskeeseen kuuluvia ensimmäistä datapakettia seuraavia datapaketteja muistista viiveen jälkeen siitä kun ajallisesti edellinen datapaketti otettiin muistista.

15. Patenttivaatimuksen 14 mukainen menetelmä, **tunnettu** siitä, että mainittu viive on näytteenottoväli.

15

16. Patenttivaatimuksen 11 mukainen menetelmä, **tunnettu** siitä, että mainittu datapurske käsittää reaaliaikaista interaktiivista dataa.

20

17. Patenttivaatimuksen 16 mukainen menetelmä, **tunnettu** siitä, että datapurske käsittää yhtä seuraavista: packet video over IP, Voice over IP, audio/video streaming over IP.

18. Patenttivaatimuksen 11 mukainen menetelmä, **tunnettu** siitä, että mainitun datapurskeen kesto on yhtäpitkä aika kuin jona lähettäjä lähettää informaatiota tauottomasti.

25

19. Patenttivaatimuksen 11 mukainen menetelmä, **tunnettu** siitä, että mainitut datapaketit vastaanotetaan tietoverkosta reaaliaikaisen yhteyden kautta johonkin toiseen tietoverkon osapuoleen.

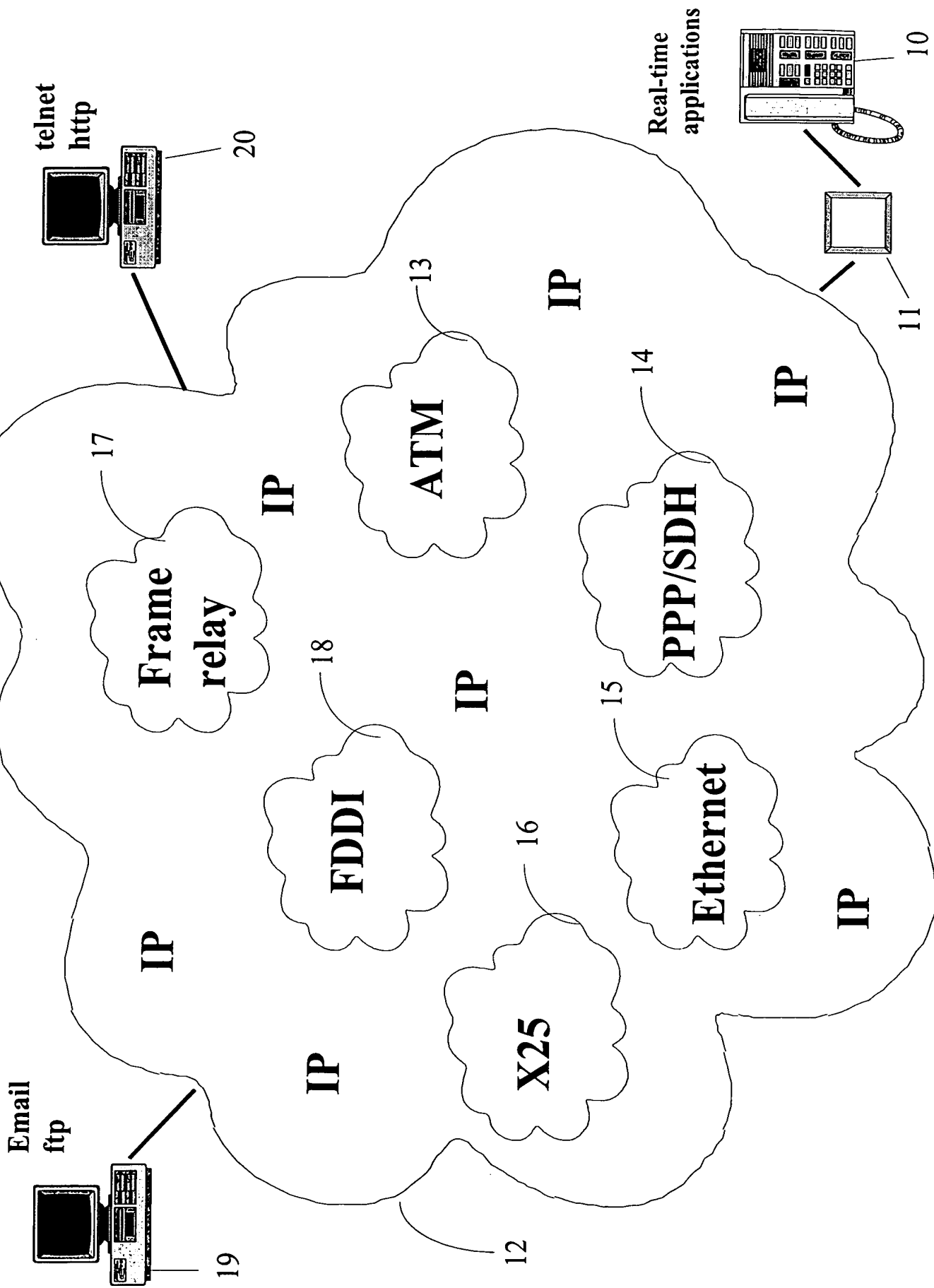
30

(57) Tiivistelmä

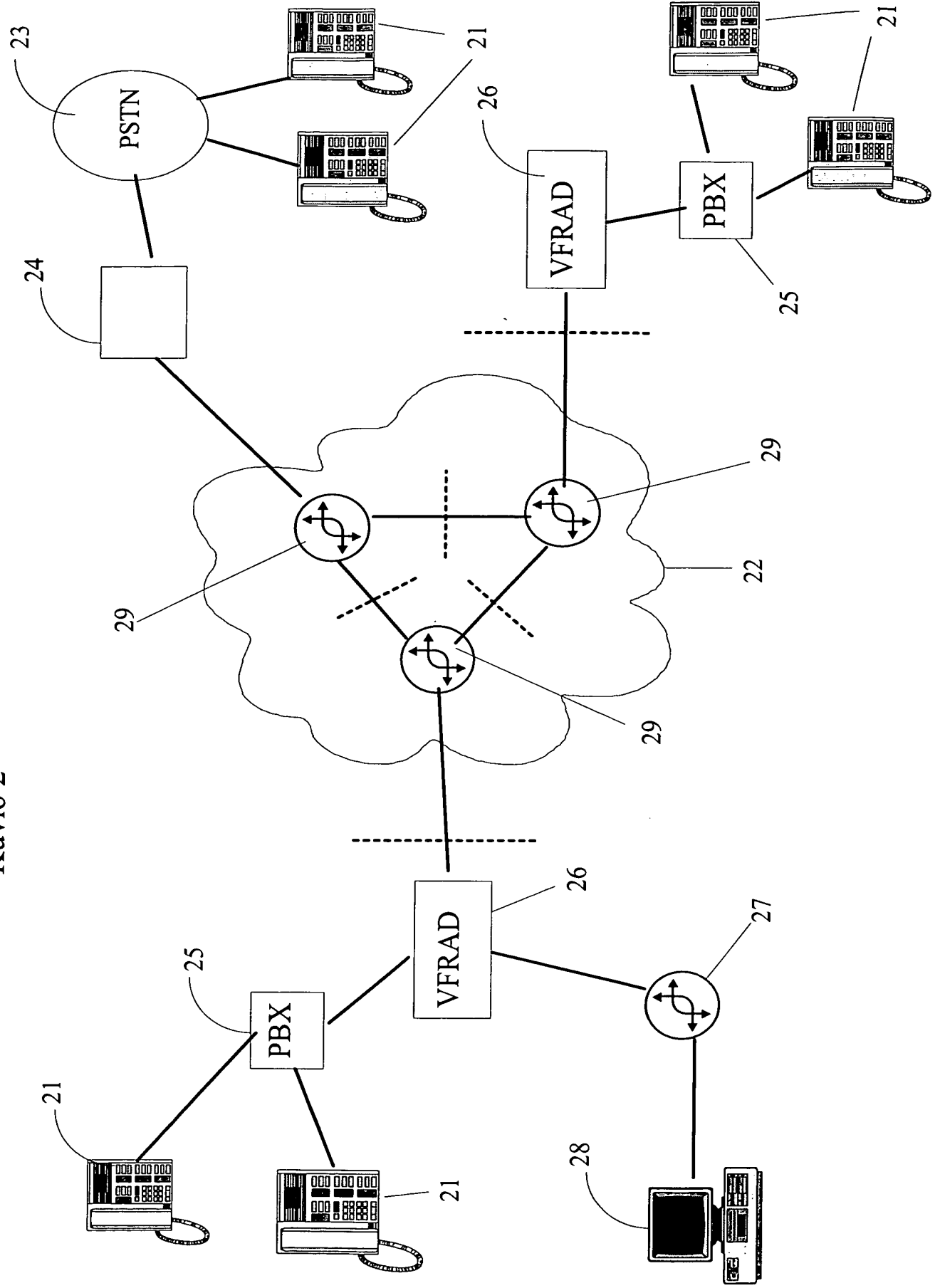
Menetelmä ja laite datapakettien prosessoinnin ajoittamiseksi, jossa menetelmässä vastaanotetaan datapurskeen osana oleva datapaketti, tallennetaan vastaanotettu datapaketti muistiin (53), otetaan datapaketti muistista toistoviiveen jälkeen siitä kun datapaketti vastaanotettiin, lasketaan toistoviiveelle arvo, jolla toistoviiveen arvolla n :stä ajallisesti viimeisestä datapaketista vain m kappaletta olisi jäänyt vastaanottamatta, jos kyseiset datapaketit käsittävien datapurskeiden prosessoinnin aloittamista olisi viivästytetty mainitun toistoviiveen verran, missä n ja m ovat luonnollisia lukuja, ja siirretään datapaketti muistista prosessointivälineille (38) kelloilta (54) saadun vasteen pohjalta mainitun toistoviiveen arvon saavuttamisesta siitä kun datapaketti vastaanotettiin.

(Kuvio 5)

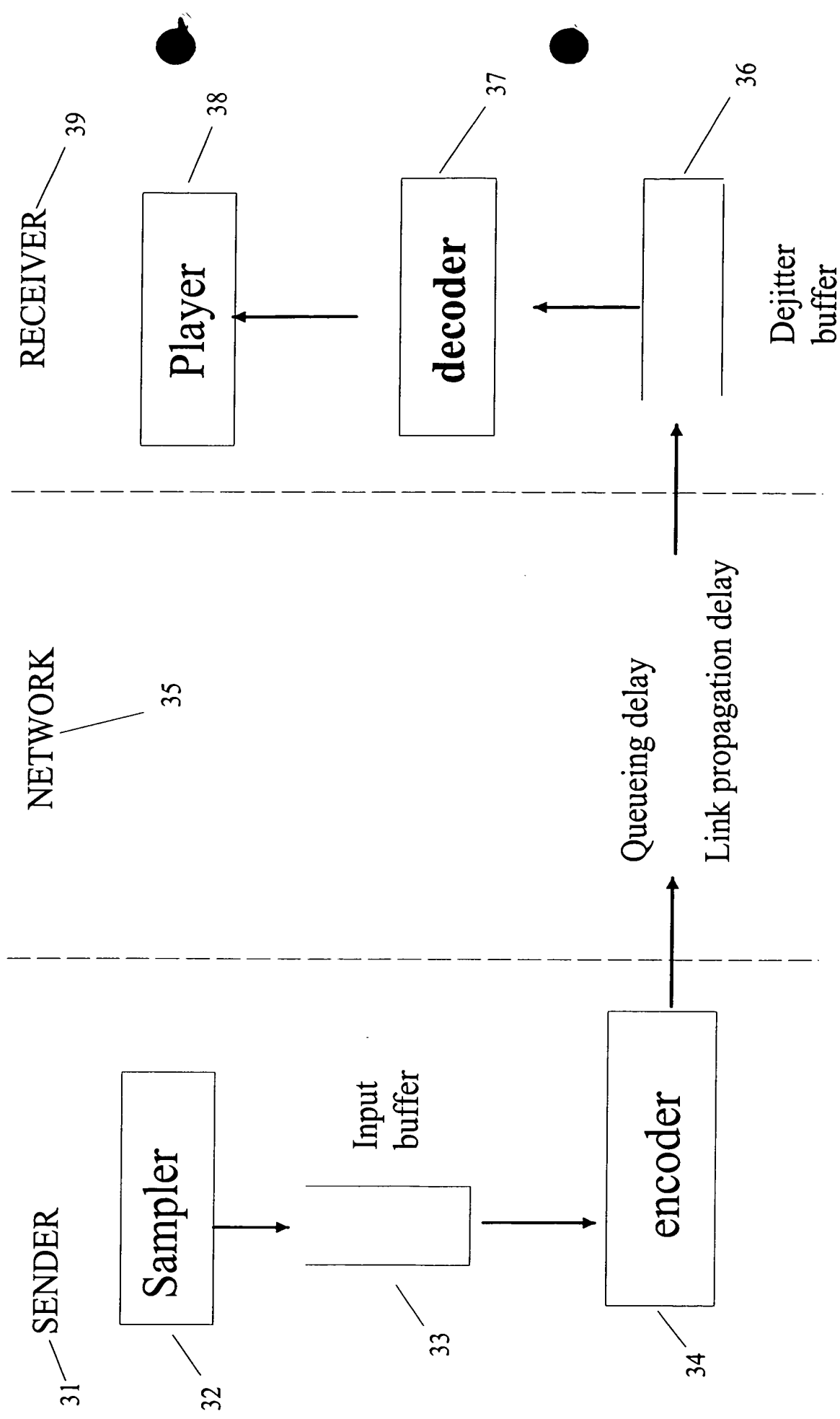
Kuvio 1



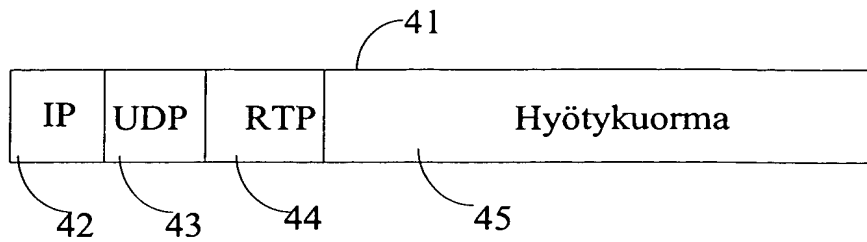
Kuvio 2



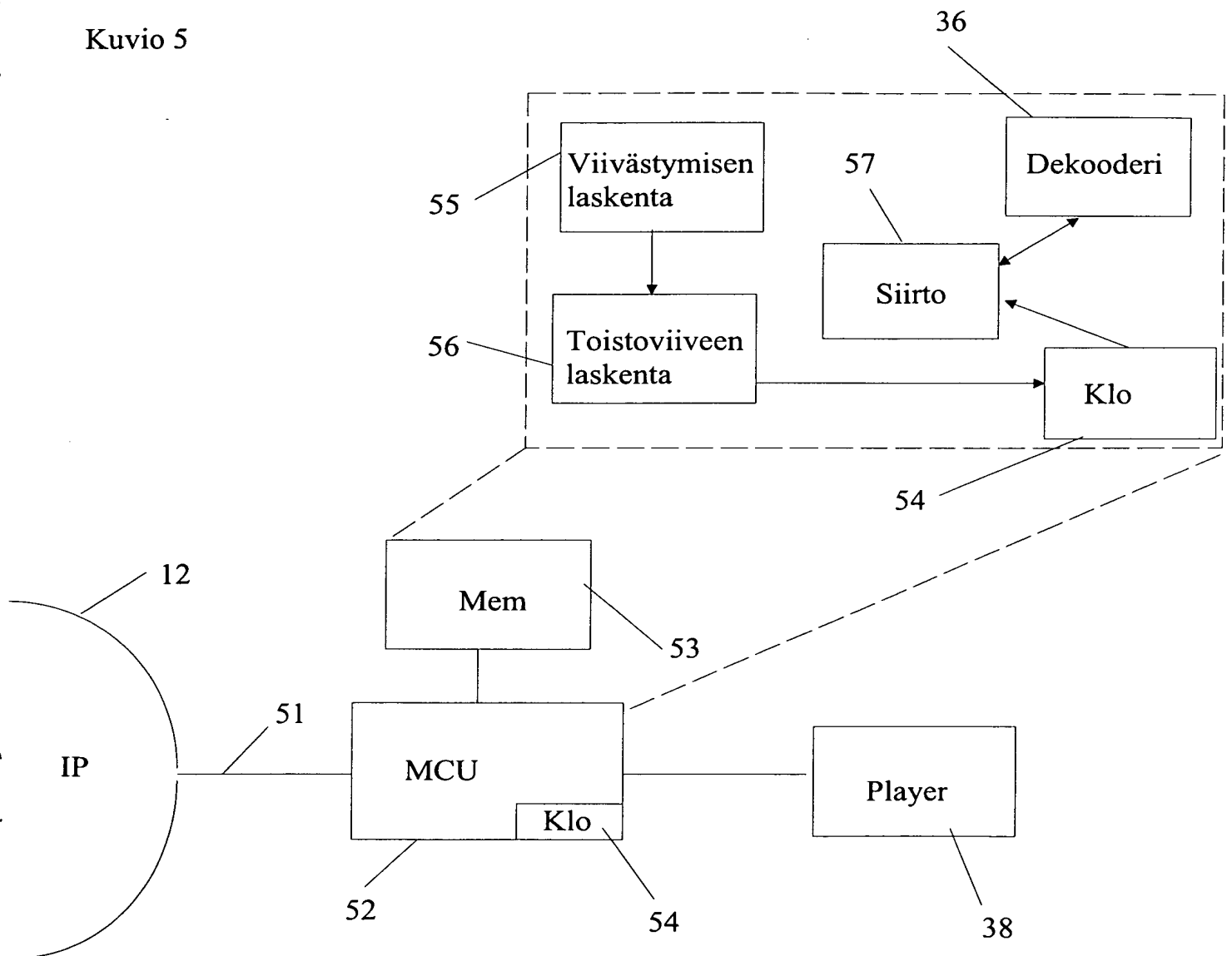
Kuvio 3



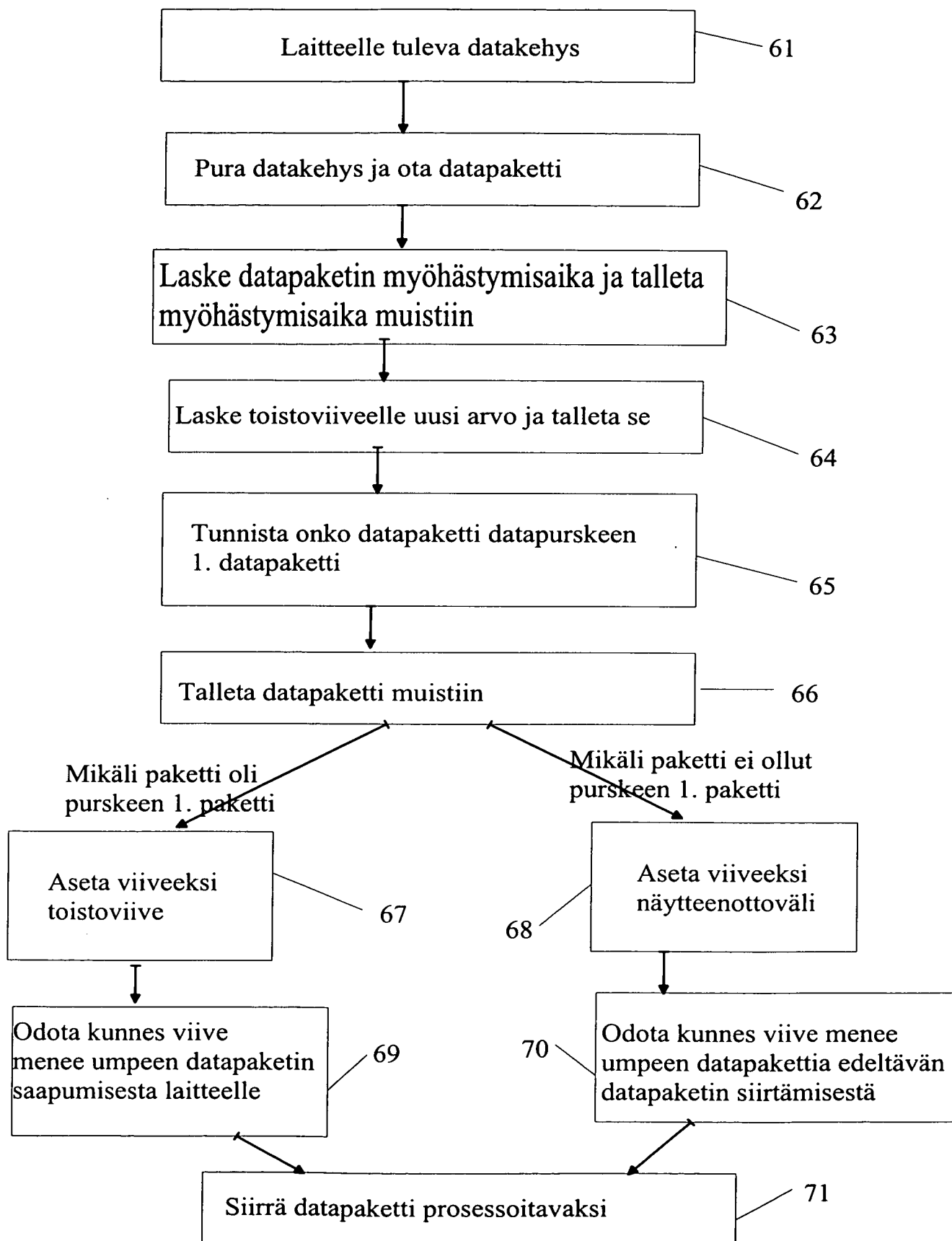
Kuvio 4

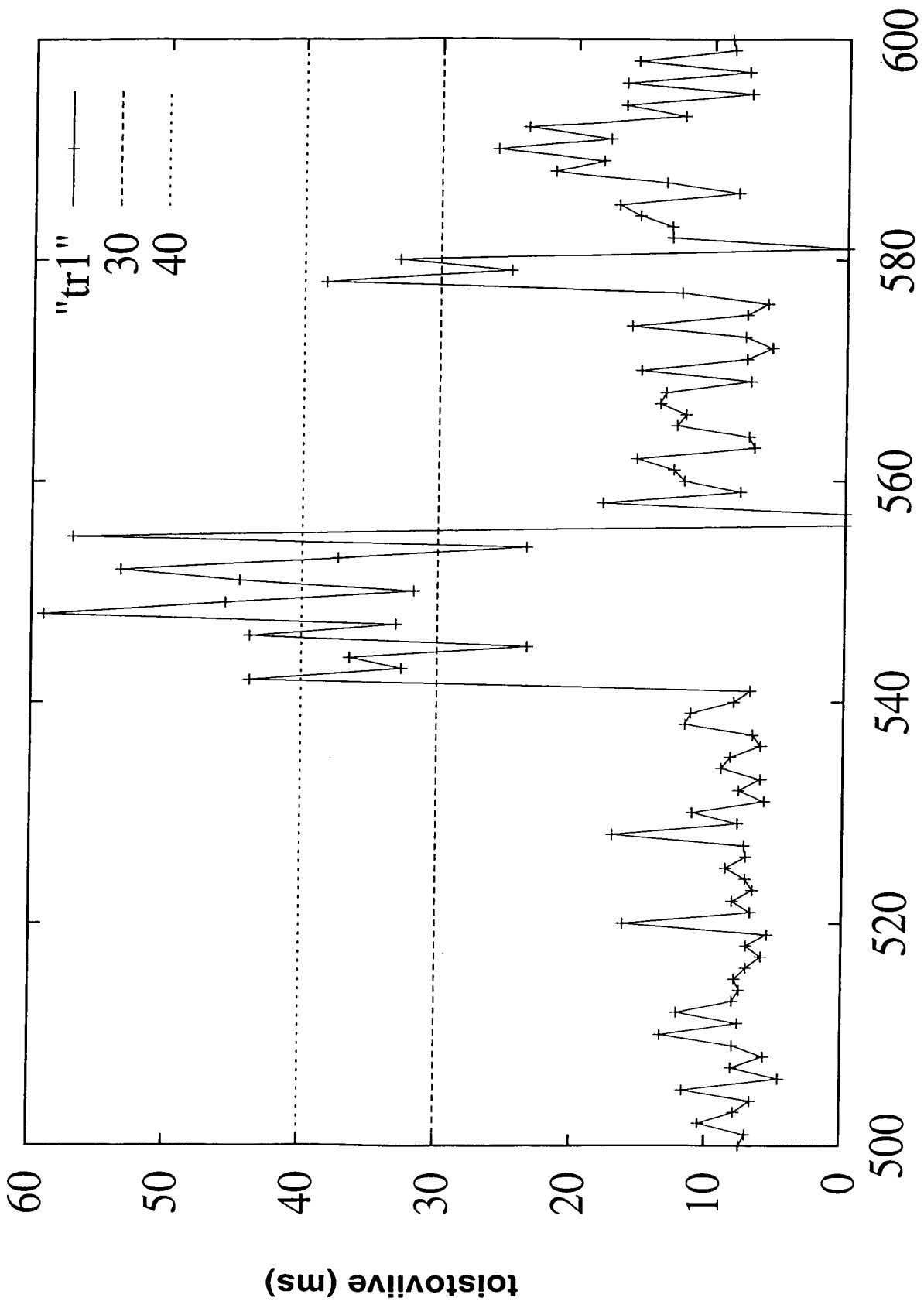


Kuvio 5



Kuvio 6





Datapaketin järjestysnumero

Kuvio 7